CLAIMS

What is claimed is:

1. A method of rotating a first image in an image buffer, the method comprising the steps of:

extracting first image data from the image buffer; and

creating a rotated image that is substantially free of aliasing error using weighted sums of data points of the first image, wherein weighting depends on a skew angle of the first image and data point location in the first image.

2. The method of claim 1, wherein the first image is of a document, and the first image data is created in the image buffer by the step of scanning the document.

3. The method of claim 1, further comprising the step of storing the first image data in a database.

4. The method of claim 1, wherein the step of creating the rotated image is provided by applying the following algorithm to the first image data:

$$V_o = K_h * K_v (V1 + V4 - V2 - V3) + K_h (V3-V4) + K_v (V2- V4) + V4,$$

wherein V_o is a data point of the rotated image; V1, V2, V3 and V4 are first image data points that each incorporate a portion of V_o; and K_h and K_v are fractions that are functions of skew angle and data point location of the first image.

5. The method of claim 4, wherein K_h and K_v are implemented in 1/8th increments.

6. The method of claim 4, further comprising the step of providing K_h and K_v in at least one lookup table.

7. The method of claim 1, wherein the step of creating the rotated image is provided by applying the following algorithm to the first image data:

$$V_o = K_h (V3-V4) + K_v (V2- V4) + V4,$$

wherein V_o is a data point of the rotated image; V2, V3 and V4 are data points of the first image that each incorporate a portion of V_o; and K_h and K_v are fractions that are functions of skew angle and data point location of the first image.

8. The method of claim 7, wherein K_h and K_v are implemented in 1/8th increments.

9. The method of claim 7, further comprising the step of providing K_h and K_v in at least one lookup table.

10. A system for rotating an initial image stored in an image buffer, the system comprising:

an image rotation module configured to rotate the initial image to create a rotated image by using weighted sums of data points of the initial image, wherein the weighting depends on a skew angle of the initial image and data point location in the initial image.

11. The system of claim 10, wherein the data points of the initial image are in adjacent rows of the image buffer.

12. The system of claim 11, wherein a pair of data points are used from each of the adjacent rows of the image buffer.

13. The system of claim 10, further comprising an image generation module configured to create the initial image.

14. The system of claim 13, further comprising a scanner for supplying data to the image generation module.

15. The system of claim 10, further comprising a database configured to store initial image data.

16. The system of claim 10, wherein the image rotation module repetitively applies the following algorithm to initial image data to create the rotated image:

$$V_o = K_h * K_v (V1 + V4 - V2 - V3) + K_h (V3-V4) + K_v (V2- V4) + V4,$$

wherein V_o is a data point of the rotated image; V1, V2, V3 and V4 are data points of the initial image that each incorporate a portion of V_o; and K_h and K_v are fractions that are functions of skew angle and data point location of the initial image.

17. The system of claim 16, wherein K_h and K_v are implemented in 1/8th increments.

18. The system of claim 16, wherein K_h and K_v are provided in a lookup table.

19. The system of claim 10, wherein the image rotation module repetitively applies the following algorithm to initial image data to create the rotated image:

$$V_o = K_h (V3-V4) + K_v (V2- V4) + V4,$$

wherein V_o is a data point of the rotated image; V2, V3 and V4 are data points of the initial image that each incorporate a portion of V_o; and K_h and K_v are fractions that are functions of skew angle and data point location of the initial image.

20. The system of claim 19, wherein K_h and K_v are implemented in 1/8th increments.

21. The system of claim 19, wherein K_h and K_v are provided in a lookup table.

22. A workstation comprising the system for rotating an initial image stored in an image buffer of claim 10.

23. A computer program product comprising a computer useable medium having computer readable program code embodied therein for processing a first image in an image buffer, the computer program product comprising:

program code configured to rotate the first image to create a rotated image by using weighted sums of data points of the first image, wherein the weighting depends on a skew angle of the initial image and data point location in the first image.

24. The program product of claim 23, wherein the program code configured to rotate the first image repetitively applies the following algorithm to first image data to create the rotated image:

$$V_o = K_h * K_v (V1 + V4 - V2 - V3) + K_h (V3-V4) + K_v (V2- V4) + V4,$$

wherein V_o is a data point of the rotated image; V1, V2, V3 and V4 are data points of the first image that each incorporate a portion of V_o; and K_h and K_v are fractions that are functions of skew angle and data point location of the first image.

25. The program product of claim 23, wherein the program code configured to rotate the first image repetitively applies the following algorithm to first image data to create the rotated image:

$$V_o = K_h (V3-V4) + K_v (V2- V4) + V4,$$

wherein V_o is a data point of the rotated image; V1, V2, V3 and V4 are data points of the first image that each incorporate a portion of V_o; and K_h and K_v are fractions that are functions of skew angle and data point location of the first image.

26. A system for rotating an initial image stored in an image buffer, the system comprising:

an image rotation module configured to rotate the initial image to create a rotated image that is substantially free of an aliasing error.

27. The program product of claim 26, wherein the program code configured to rotate the first image repetitively applies the following algorithm to initial image data to create the rotated image:

$$V_o = K_h * K_v (V1 + V4 - V2 - V3) + K_h (V3-V4) + K_v (V2- V4) + V4,$$

wherein V_o is a data point of the rotated image; V1, V2, V3 and V4 are data points of the initial image that each incorporate a portion of V_o; and K_h and K_v are fractions that are functions of skew angle and data point location of the initial image.